BGP SECURE ROUTING EXTENSION (SRX)
QUICK USERS GUIDE

Introduction

BGP SRx in an addition to the BGP routing infrastructure. It provides an API to router implementations that wish to use origin validation as well as path validation. The BGP-SRx will be used as a validation service. This keeps the impact (software/memory/processing) on the router to a minimum. SRx provides an API that allows embedding a small proxy into the router. This proxy handles the router proxy communication.
This design allows the SRx server to serve multiple BGP router instances.

State of BGP-SRx

BGP-SRx is NOT intended to be used in a production system. BGP-SRx is still in the stage of a prototype and might face some instability. In such a case we appreciate every input that helps to improve the stability as well as performance. The code is open source. Feedback can be send to bgpsrx-dev@nist.gov.

Installation

BGP-SRx provides configuration scripts. To install SRx, download the package from http://www-x.antd.nist.gov/bgpsrx. Once downloaded deflate the package and call the configuration method. The INSTALL contains an example on how to configure the service. This package also generates the SRx-API that is needed for QuaggaSRx. Follow the configuration messages and install missing libraries as needed. Under fedora all libraries can be installed using yum. The internal prefix tree (Patricia) tree is bundled in this package. IF you decide to use the default one, check the patch file located in srx/extras and update the installed version. Recompilation is necessary. If not, use the switch –with-patr and BGP-SRx will be compiled with the bundled version.
Once compiled call make; make install. To select an individual installation directory use the configuration parameter --prefix. This directory is needed for QuaggaSRx to specify the location of the API binaries.

The Server Software [srx_server]

The BGP-SRx Server is the main component the collects ROA information from the validation of the Validation Cache using the router to cache protocol. The BGP router connects to SRx using the SRx-API and sends update validation requests. The requests are split into an origin validation and a path validation request. The SRx server answers by sending the validation result for both separate, the origin

validation and path validation. Once the SRx server recognizes a change in the validation state it sends a notification to the routers.
The SRx-server can be accessed using a telnet session.

Available commands are:

close, quit, exit	Close this console!
shutdown <password>	Shutdown the SRx Server!
log-level <number>	Set the log level of the server. 3=ERROR, 5=NOTICE, 6=INFO, 7=DEBUG
rtr-sync [proxyID]	Send synchronization request to the provided proxy or all.
rtr-goodbye [proxyID]	Close the connection to provided proxy or all!
show-srxconfig	Display the configuration of the srx server
show-update <cmd>	Display update data according to the command string.
cmd:= id <id>	Show the update with the ID (hex).
num-updates	Display the number of updates stored in update cache!
num-prefixes	Display the number of prefixes stored in the prefix cache!
num-proxies	Display the number of proxies attached
command-queue	Displays the content of the command queue.
dump-pcache	Dump the prefix cache to command line of SRx! (WARNING: this command dumps the complete SRx cache on the command line. This function should only be used for debugging of small data sets!)
!! [<parameter>]	Repeat last command with optional new parameter if specified, otherwise old parameter!

Configuration Settings:

The configuration of the SRx server can be provided using a configuration file and command lines parameters. The configuration file is called srx_server.conf and must be located next to the server binary.

```
# print information on the command console
# verbose  = true|false;
verbose = true

# specifies the log level for output (3=ERROR, 5=NOTICE, 6=INFO, 7=DEBUG)
loglevel = 3;

# specify the log file name, otherwise log information will be send to the
# console.
log    = /var/log/srx_server.log;

# if enabled the SRx server will send a synchronization request to the router.
# It is expected that the router will send validation requests for all updates in
# its tables to SRx server.
```

```
# sync = true|false;
sync    = true;

# The port address the SRx server is listening on for connections from the
# router.
port = 17900;

# Console configuration:
console: {
        # Port address of the server console.
        port = 17901;
        # The password used to shutdown the BGP-SRx server
        password = "x";
};

# Configuration for Validation Cache:
rpki: {
        # Server address of the validation cache
        host = "localhost";
        # Port address of the validation cache. Protocol: router to cache
        port = 50001;
};

# Configuration for BGPSEC data server: This is one of the hocks for future
# BGPSEC validation. The setting is mandatory!
bgpsec: {
        host = "localhost";
        port = 50002;
};
```

Each of the settings can be specified as command line setting. In case no configuration file exists each of the above settings MUST be provided as command line attribute. The command line attributes are specified with leading "--" and the configuration attribute. Attributes locates within a section must be specifies as "--section.attribute".
IMPORTANT: Command line parameters overrule configuration script parameters.

Tools

This software package comes with a set of tools used to test SRx. This tools simulate a validation cache, a validation cache client as well as a router / srx-client.

srxsvr_client

This tool is an example implementation of the BGP-SRx server client. It helps to test functions of the SRx server without the need of a full-blown API implementation such as QuaggaSRx. This tool provides a command line console and a set of

commands to connect/disconnect to a BGP-SRx server, to send validation requests etc. It implements most of the API and can be used as an example implementation for someone who wants to use the BGP-SRx API.
The Help command provides a list of commands this client can send. Also most commands with parameters can be used with default values. For example the connect command, entered with an incomplete set of attributes will request the necessary parameters and provides default values.

rpkirtr_svr

The tool is a Validation Cache simulator. It provides the validation cache interface to BGP-SRx and can be used to inject ROA information to the system. The command line console can be used to add and delete ROA entries. It also allows loading ROA information via a script. Refer to the example files provided in this distribution.
This tool too provides a Help command.

rpkirtr_client

This tool allows to test a validation cache. In this case it can be used as tester for the rpkirtr_svr tool. It helps to debug the rpki cache test harness without the need of a full-blown BGP-SRx instance.